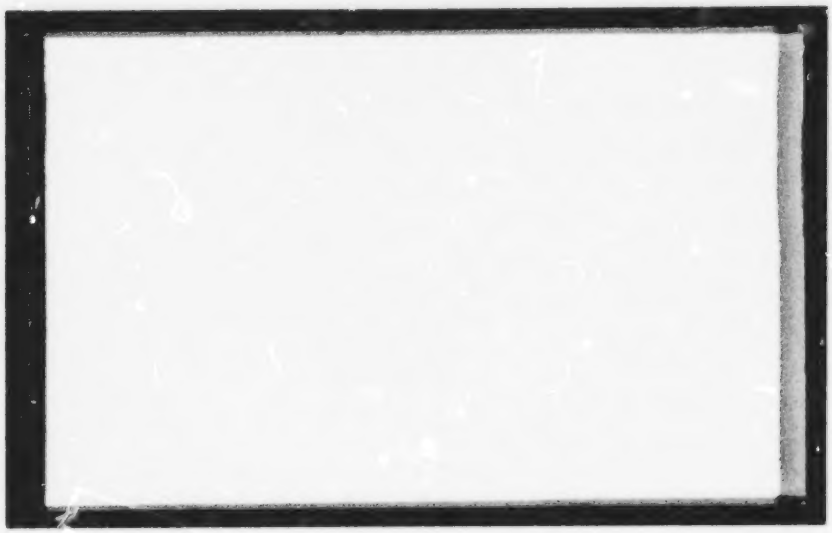
UNIVERSITÉ DE MONTRÉAL



CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, STATION A, MONTRÉAL, QUÉBEC, H3C 3J7

AN EXACT ALGORITHM FOR MINIMIZING DELIVERY AND OPERATING COSTS IN DEPOT LOCATION

by

Gilbert Laporte

and

Yves Nobert

Université de Montréal

Centre de recherche sur les transports - Publication # 146

Ecole des Hautes Etudes Commerciales - Rapport de recherche # 79-03

Janvier 1979

Dépôt légal - 4e trimestre 1979

ABSTRACT

The problem of locating a single depot among n points is considered. The objective is to minimize the sum of depot operating cost and delivery cost. The best depot location is found by means of an exact algorithm that determines simultaneously both the best depot location and the associated optimal delivery routes. A global integer programming formulation of the problem is given; the model is solved by relaxing most of its constraints and by introducing them only when they are violated. Two versions of the algorithm have been applied successfully to random problems ranging from 20 to 50 points.

1. INTRODUCTION

In recent years, many locational decisions studies in the literature have been devoted to the optimal location of one or many depots (or plants) in order to minimize

(1) the weighted sum of the arcs from the depot(s) to a set of points (customers) (see for example [4,12,17,19,20])

or

(2) the maximum (weighted) arc from the depot to the points (see for example [3,3,21]).

Note : For further references, the reader should consult the extensive bibliographies compiled by Francis and Goldstein [2] and by Lea [9,10].)

The first of these models appears to be quite suitable when trips are made from the points to the depot or when it is impossible to visit more than one point in a single journey from the depot. The second model deals with worst case situations and has been mostly applied to the optimal location of emergency facilities.

Neither of these two models takes into account the fact that a truck leaving the depot could call at more than one customer in a single visit. This aspect of the problem was recognized by Wren and Holliday [23]: "In practice the customers may be allocated to individual depots, where necessary, by the solution of a standard transportation problem. This is however, unrealistic, since the costs are not proportional to the number of units delivered to a customer, and the decision as to which depot should supply a particular customer should depend on the relative position of neighbouring customers and on the best depots to serve these customers and other which may conveniently be linked with them." The close relationship between optimal routing and depot siting was considered by Sussams [22]: "The best set of vehicle routes to serve a given area is not necessarily that which minimizes the distance travelled. Efficient routing is inextricably bound up with depot siting", and by Rand [18] : "Many practioners

are aware of the danger of suboptimizing by separating depot location from vehicle routing." Rand appreciates the complexity of the problem: "Unfortunately, it will rarely be practicable to determine depot locations using vehicle scheduling packages because of the tremendous increase in computer time."

When a set of n customers have to be supplied from the same depot by the same delivery van, the optimal delivery cost will be the same for all depots (the total length of the optimal travelling salesman tour) and the depot location problem is of course non-existent. However, if deliveries are to be made by m salesmen located at one or many depots, the cost function will then be affected by the choice of the depot(s). In the case where there is only one depot, one could of course solve optimally n times an m-travelling salesmen problem (m-TSP) as was suggested by Lawrence and Pengilly [8]. However, the best depot location (corresponding to the optimal m-TSP solution) can be found somewhat less clumsily by means of an algorithm that determines simultaneously both the best depot location and the optimal m-TSP tour.

In this paper, a global integer programming formulation of the problem and an optimal algorithm are presented. Two versions of the algorithm have been applied successfully to random problems ranging from 20 to 50 cities.

2. FORMULATION

The formulation of the problem is similar to that of the m-TSP [6,7]. Let us consider n points to be visited by m (> 1) salesmen. (m is a known integer.) Let $D=(d_{ij})$ be a distance matrix between the points. For the sake of brevity, we will only consider the case where D is symmetrical although the extension of our formulation to the asymmetrical case is straightforward. D need not be euclidean. Let us define the variables

as follows: to each (undirected) arc between two points i and j, we associate a variable x_{ij} which indicates the number of times (0,1 or 2) the arc is used by a salesman. To each node k corresponds a binary variable y_k which takes the value 1 if the node is used as a depot and the value 0 otherwise. Let f_k be the cost of operating a depot at node k. The integer linear programming formulation is as follows.

$$\text{Minimize} \quad \sum_{i<j} \sum d_{ij} x_{ij} + \sum_{k=1}^{n} f_k y_k$$

subject to

$$(1) \quad \sum_{k=1}^{n} y_k = 1$$

$$(2) \quad \sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2 + 2(m-1)y_k \qquad (k = 1,\ldots,n)$$

$$(3) \quad \sum_{\substack{i<j \\ i,j\in S}} x_{ij} \leq |S| - 1 + (m-1) \sum_{k\in S} y_k$$

(for all $S \subseteq \{1,\ldots,n\}$ and $2 \leq |S| \leq n-2$)

$$(4) \quad 0 \leq x_{ij} \leq 2 \qquad (i < j)$$

$$(5) \quad 0 \leq y_k \leq 1 \qquad (k = 1,\ldots,n)$$

$$(6) \quad x_{ij},\ y_k \text{ integer} \qquad (i < j;\ k = 1,\ldots,n)$$

The objective function includes a travel cost and a depot operating cost. In practice, these two costs should be scaled so that they relate to the same time period (e.g. one year).

Constraint (1) merely indicates that only one city should be used as a depot. Hence all m salesmen are based at the same depot. It is possible in principle to solve problems with $p > 1$ depots although this would add to the complexity of the model.

Constraints (2) state that the degree of a node must be equal to 2m if the node is used as a depot and to 2 otherwise.

Constraints (3) ensure that the solution contains no illegal subtour. These occur whenever none of the nodes included in the subtour is used as a depot. If a subset S of {1,...,n} does not contain the depot or if m=1, constraints (3) become

$$\sum_{\substack{i<j \\ i,j\in S}} x_{ij} \leq |S| - 1 \qquad (2 \leq |S| \leq n-2)$$

i.e. the same as in the 1-TSP [14,15,16].

If S contains the depot,

$$(|S| - 1 + m \sum_{k\in S} y_k)$$

is an upper bound on the number of arcs contained in S and used by a salesman. The computation of this bound assumes that all m-salesmen tours only involve points in S. However, as long as $|S| \leq n-2$, this upper bound cannot be reached as there would be no salesman left for points outside S. Hence

$$(|S| - 1 + (m-1) \sum_{k\in S} y_k)$$

is a valid upper bound.

It is worth noticing that the formulation reduces to that of the m-TSP when the depot is fixed. In this case, one of the y_k's is fixed to 1, ($y_1 = 1$, say) constraint (1) can be eliminated, constraints (2) become

$$\sum_{j=2}^{n} x_{1j} = 2m$$

and

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2 \qquad (k = 2,\ldots,n)$$

and constraints (3) become

$$\sum_{\substack{i<j \\ i,j\in S}} x_{ij} \leq |S| - 1 \qquad (2 \leq |S| \leq n-2 \text{ and } 1 \notin S)$$

3. ALGORITHM

Before solving the model as it is formulated above, an upper bound on the objective function is found by applying the following simple heuristic procedure to each node i:

1) fix the depot at node i;
2) transform the associated m-TSP into a 1-TSP as suggested in [11], for example;
3) obtain an upper bound on the transformed problem by means of a 2-opt algorithm [12].

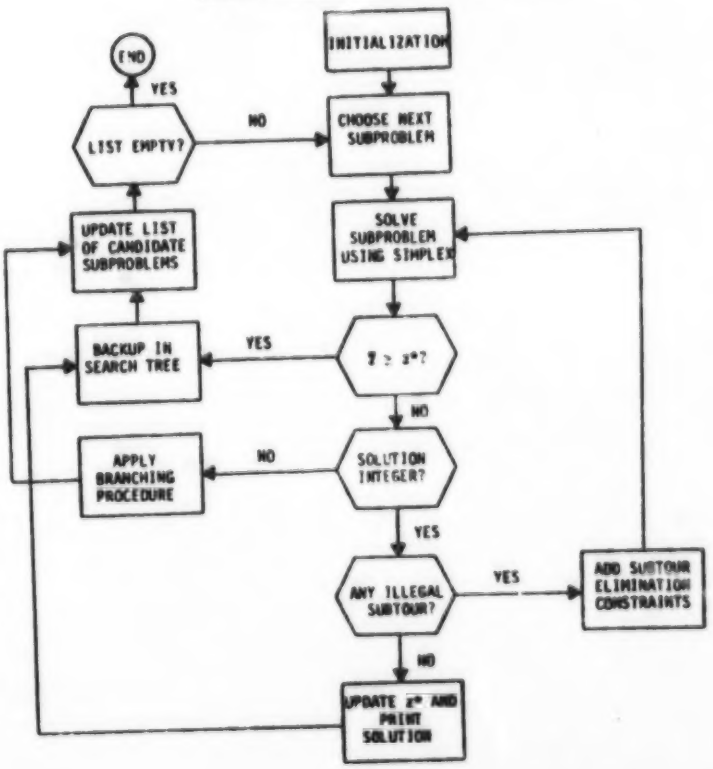
The upper bound is then chosen as the smallest of the n bounds.

It is obvious that the problem cannot be solved directly by using the above formulation as the number of constraints would become excessive, even for relatively small values of n. Some form of constraint relaxation is needed. In our approach, we have chosen to solve the problem by first relaxing constraints (3) and (6). Miliotis' previous experience [14,15] with the 1-TSP has shown that it is inappropriate to relax constraints (2) as most of these constraints tend to be effective in the optimal solution. Subtour elimination constraints (3) are introduced only when an integer solution containing one or more illegal subtours has been found.

Integrity (constraints (6)) is obtained by a branch and bound algorithm [5]. As in most problems of this kind, the same result can also be achieved by a cutting planes (Gomory cuts) algorithm. Although, cutting planes have proved very efficient in similar problems [6,15,16], they have failed to converge to a feasible solution in most depot location problems we have attempted. The presence of the y_k variables in this problem means that the coefficient structure of the problem is no longer "nearly unimodular" as was the case for the 1-TSP and for the m-TSP. The use of cutting planes for this problem has therefore been rejected. The algorithm can be summarized by the flowchart drawn in figure 1.

FIGURE 1

FLOWCHART OF THE ALGORITHM



NOTE : z^* = cost of best solution so far
$\bar{z}$ = cost of current subproblem
$\bar{z}$ = ∞ whenever subproblem is infeasible

We have experimented with two branching strategies for this problem. The first strategy consists in selecting the (fractional) variable which would produce the largest increase in the value of the objective function if it was branched upon in a given direction, and then in branching in the opposite direction. This is fairly standard procedure and it is used by Land and Powell [5]. In the second strategy, the (fractional) y_k variable which would produce the smallest increase in the objective function is fixed at 1 as soon as possible in the branching process. If there is no fractional y_k variable, the first branching strategy applies. The rationale for this rule is that by so doing, one reduces many branches of the search tree to a simple m-TSP at an early stage; moreover, if the variable y_k that is fixed to 1 is chosen carefully, the optimal solution (and hence a good bound) can be found quickly. Computational results with both strategies are presented in the next section. Figures 2 and 3 show the search tree obtained by each strategy on a euclidean 20 city, 3-TSP.

FIGURE 2

FIRST BRANCHING STRATEGY

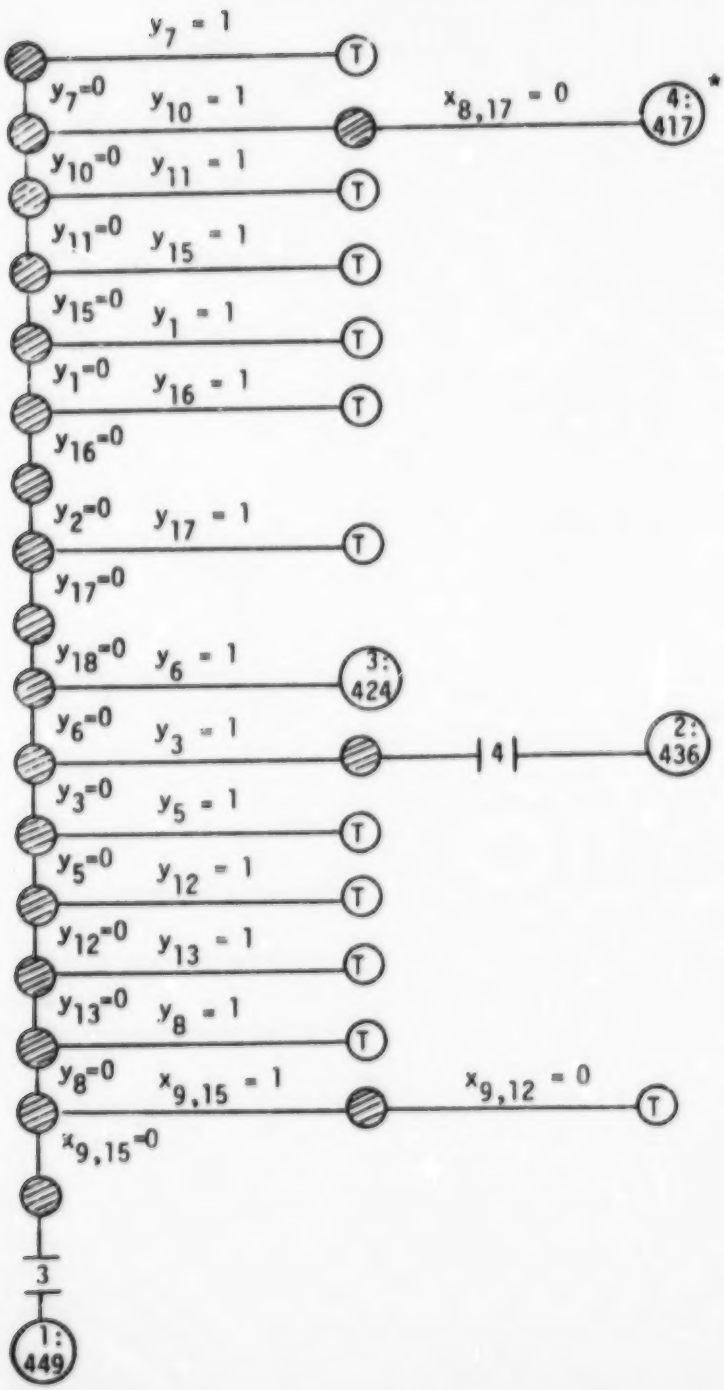
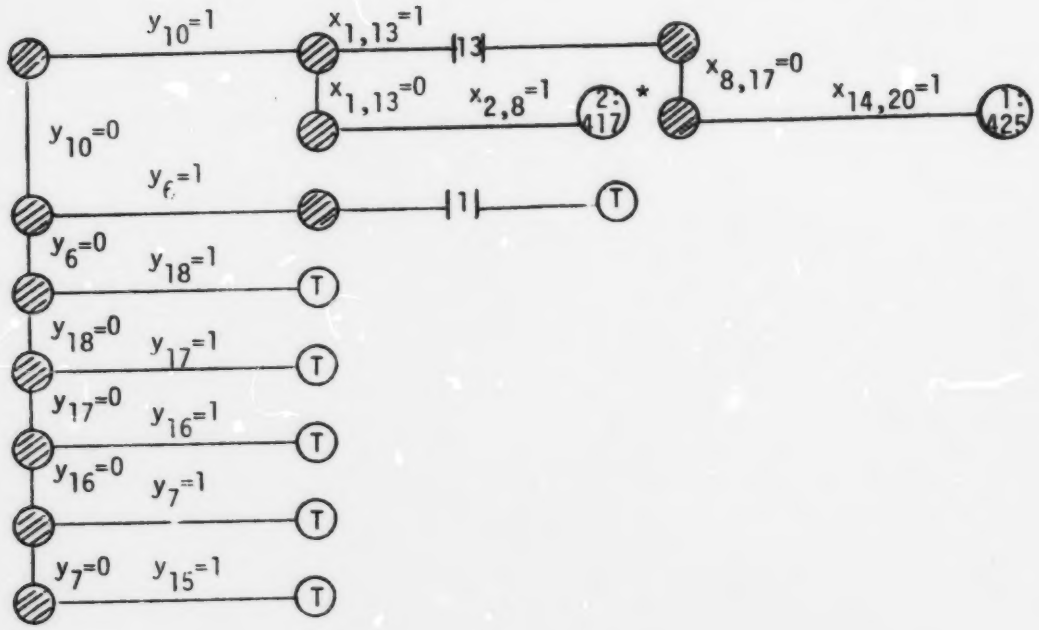


FIGURE 3

SECOND BRANCHING STRATEGY



NOTATION FOR FIGURES 2 AND 3 :

- fractional solution
- T — a tail fo the tree
- p — p subtour elimination constraints
- i: z — feasible solution number i with cost equal to z
- * — optimal solution

4. COMPUTATIONAL RESULTS

In all tests that follow, solution times are in number of seconds. The computer used was the University of Montreal CYBER 173. Moreover, in each case, the graph is complete, i.e. all $\frac{1}{2}n(n-1)$ arcs between pairs of points exist.

The object of our first test was to illustrate that it is preferable to use our algorithm than to solve separately n problems each having a fixed depot. Table 1 shows the results for 20 city problems solved by considering in turn each node as a depot.

TABLE 1

Node used as depot	Euclidean problem (m = 3)		Non euclidean problem (m = 5)	
	Solution time	Optimal cost	Solution time	Optimal cost
1	.94	342	1.04	339
2	1.22	367	.78	253
3	1.06	336	1.16	401
4	.94	380	.91	371
5	1.23	358	.95	320
6	1.40	427	1.20	275
7	1.31	400	1.17	367
8	1.00	359	1.02	390
9	1.28	351	.94	460
10	1.20	360	1.80	403
11	.98	396	.87	241
12	1.96	364	.80	330
13	1.37	345	1.23	234*(opt)
14	1.24	350	.92	311
15	1.50	359	1.55	565
16	1.15	353	.83	253
17	1.56	347	.92	315
18	1.10	332*(opt)	.88	346
19	1.53	354	1.16	383
20	1.40	357	1.83	351
	Total 25.37		Total 21.86	

In each of these problems, the times taken by our algorithm to solve the problem were 4.71 and 10.5 seconds respectively (first branching strategy). This compares well with the total times of 25.37 and 21.86 seconds required to solve 20 m-salesmen problems. Moreover, results from a previous study on the m-TSP [6] and tables 2 to 5 below suggest that the gains made by using our algorithm should increase as n becomes larger. Another interesting feature illustrated by table 1 is that the value of the optimal solution varies widely according to the node chosen as a depot. This confirms the need to have an algorithm which takes simultaneously into account both the depot location and the optimal routes.

Tables 2 to 5 give the results for a set of problems ranging from 20 to 50 cities. The two branching strategies are compared, both on euclidean and non euclidean problems. In order to make comparisons with the m-TSP easier, all f_k's have been set to 0. As a rule, it may be said that the second branching strategy appears to be more efficient, especially in larger problems. Relatively few type (3) constraints (subtour elimination constraints) are needed and this explains, to a large extent, the success of our algorithm. The depot chosen by the heuristic procedure at the beginning of the algorithm rarely corresponds to the best depot given by our algorithm.

TABLE 2

EUCLIDEAN PROBLEMS, FIRST BRANCHING STRATEGY

n	problem number	m	maximum size of the inverse (*)	number of constraints	number of subtour constraints	number of simplex iterations	solution time	optimum (depot)	heuristic solution (heuristic depot)
20	1	3	30	34	13	127	4.71	332(18)	345(3)
	2	3	35	47	26	545	27.96	410(8)	443(13)
	3	3	32	38	17	359	14.07	417(10)	474(9)
	4	3	33	44	23	447	17.78	393(19)	446(4)
	5	5	32	37	16	666	22.33	438(10)	485(5)
25	1	4	36	50	24	464	25.04	471(22)	540(4)
	2	3	37	47	21	172	9.98	440(14)	476(15)
	3	3	42	46	20	358	19.12	407(13)	450(21)
	4	3	40	49	23	313	19.23	395(14)	420(25)
	5	4	41	53	27	1113	70.96	483(8)	539(4)
30	1	2	49	72	41	429	48.59	435(30)	451(9)
	2	4	46	50	19	525	35.71	445(4)	503(17)
	3	4	50	74	43	1422	138.26	487(3)	522(24)
	4	4	46	62	31	797	61.46	472(10)	501(7)
	5	4	44	52	21	1459	137.15	487(17)	487(17)
35	1	4	56	78	42	2625	386.75	541(8)	581(29)
	2	5	52	61	25	1083	104	486(27)	548(2)
	3	4	53	63	28	894	84.51	482(7)	560(32)
	4	3	52	55	19	412	41.26	486(8)	570(28)
	5	3	58	69	33	946	119.82	468(5)	524(29)
40	1	3	62	82	41	736	140.4	469(6)	537(5)
	2	3	68	81	40	3392	694.62	542(4)	593(23)
50	1	4	80	101	50	2596	935.83	565(10)	616(49)
	2	3	81	112	61	2128	688.34	505(26)	556(15)

(*) This column merely refers to the maximum number of effective constraints in the program. For further informations the readers are refered to [5].

TABLE 3

EUCLIDEAN PROBLEMS, SECOND BRANCHING STRATEGY

n	problem number	m	maximum size of the inverse	number of constraints	number of subtour constraints	number of simplex iterations	solution time	optimum (depot)	heuristic solution (heuristic depot)
20	1	3	30	33	12	278	9.35	332(18)	345(3)
	2	3	34	44	23	422	20.21	410(8)	443(13)
	3	3	30	35	14	258	9.13	417(10)	474(9)
	4	3	29	36	15	241	9.79	292(19)	446(4)
	5	5	32	36	15	907	30.61	438(10)	485(5)
25	1	4	37	40	14	264	11.34	471(22)	540(14)
	2	3	38	44	18	237	12.18	440(14)	476(15)
	3	3	41	45	19	209	10.51	407(15)	450(21)
	4	3	39	45	19	220	11.64	395(14)	420(25)
	5	4	43	49	23	792	43.69	483(8)	539(4)
30	1	2	49	73	42	365	37.84	435(9)	451(9)
	2	4	45	50	19	1166	78.89	445(4)	503(17)
	3	4	51	65	34	1572	143.07	487(3)	522(24)
	4	4	54	68	37	11180	979.85	472(10)	501(7)
	5	4	45	52	21	1224	83.86	487(17)	487(17)
35	1	4	56	76	40	11394	1097.44	541(8)	581(29)
	2	5	55	62	26	2009	217.11	486(27)	548(2)
	3	4	55	60	24	1181	110.34	482(7)	560(32)
	4	3	55	66	30	598	56.43	486(8)	570(28)
	5	3	55	65	29	478	55.84	468(5)	524(29)
40	1	3	63	86	45	1687	355.34	469(6)	537(5)
	2	3	*	*	*	9736	*	565(34)	593(23)
50	1	4	80	99	48	2280	616.5	565(10)	616(49)
	2	3	88	102	51	2991	726.78	505(44)	556(15)

(*) Time limit.

TABLE 4

NON EUCLIDEAN PROBLEMS, FIRST BRANCHING STRATEGY

n	problem number	m	maximum size of the inverse	number of constraints	number of subtour constraints	number of simplex iterations	solution time	optimum (depot)	heuristic solution (heuristic depot)
20	1	5	27	37	16	345	10.5	234(13)	254(13)
	2	2	26	28	7	62	1.79	207(14)	291(5)
	3	5	28	39	18	398	12.83	260(7)	359(15)
	4	2	28	34	13	46	1.31	185(5)	313(5)
	5	3	30	34	13	240	10.5	179(2)	276(18)
25	1	2	32	37	11	49	1.97	213(10)	300(25)
	2	5	36	45	19	952	45.06	203(16)	277(24)
	3	3	33	40	14	99	4.39	180(12)	258(23)
	4	3	37	46	20	338	20.72	193(2)	247(24)
	5	3	35	42	16	492	26.62	219(20)	253(20)
30	1	2	43	49	18	132	8.15	197(26)	239(20)
	2	3	43	50	19	547	39.5	195(8)	286(18)
	3	4	41	47	16	740	52.34	235(17)	280(17)
	4	4	42	53	22	636	46.88	253(19)	293(19)
	5	5	47	67	36	1640	125.4	253(8)	341(4)
35	1	4	53	80	44	2270	263.48	192(17)	289(32)
	2	3	51	60	24	536	51.25	173(7)	243(15)
	3	4	47	55	19	357	32.33	183(7)	307(7)
	4	4	52	73	37	2806	333.76	252(29)	392(2)
	5	4	50	60	24	1020	114.58	211(5)	292(2)
40	1	3	56	74	33	932	109.87	243(13)	365(24)
	2	4	59	73	32	1496	197.89	219(22)	329(2)
50	1	1	*	*	*	*	*		
	2	2	*	*	*	*	*		

(*) Time limit

TABLE 5

NON EUCLIDEAN PROBLEM, SECOND BRANCHING STRATEGY

n	problem number	m	maximum size of the inverse	number of constraints	number of subtour constraints	number of simplex iterations	solution time	optimum (depot)	heuristic solution (heuristic depot)
20	1	5	28	39	18	589	19.27	234(13)	254(13)
	2	2	25	28	7	54	1.56	207(19)	291(5)
	3	5	26	28	7	639	19.21	260(7)	359(15)
	4	2	28	34	13	46	1.33	185(5)	313(5)
	5	3	30	34	13	268	10.53	179(2)	276(18)
25	1	2	32	37	11	49	1.98	213(10)	300(25)
	2	5	35	37	11	1109	51.66	203(16)	277(24)
	3	3	33	40	14	70	3.34	180(12)	252(23)
	4	3	38	51	25	445	25.28	193(2)	247(24)
	5	3	34	39	13	130	6.14	219(20)	253(20)
30	1	2	45	50	19	122	8.63	197(26)	239(20)
	2	3	43	52	21	215	14.06	195(8)	286(18)
	3	4	43	47	16	841	54.56	235(17)	280(17)
	4	4	45	56	25	844	61.9	253(19)	293(19)
	5	5	42	46	15	1126	75.19	253(8)	341(4)
35	1	4	52	72	36	1146	125.52	192(17)	289(32)
	2	3	51	60	24	307	28.08	173(7)	243(15)
	3	4	53	70	34	1505	160.49	183(7)	307(7)
	4	4	57	69	33	1926	182.83	252(13)	392(2)
	5	4	51	60	24	1211	117.67	211(5)	292(2)
40	1	3	55	66	25	616	69.24	243(18)	365(24)
	2	4	56	70	29	559	75.07	219(22)	329(2)
50	1	4	74	89	48	1650	321.49	203(35)	355(38)
	2	4	74	88	47	2899	546.12	237(49)	287(34)

All results presented in tables 1 to 5 assume there are no fixed costs f_k attached to the depots. We felt that as long as the f_k's were chosen within the same range as the d_{ij}'s, their presence should not affect in any significant way the performance of the algorithm. Furthermore, it is easier to draw conclusions on the speed of the algorithm when fewer problem characteristics are allowed to vary. We nevertheless solved two sets of problems with positive f_k's: ten 20-city problems (5 euclidean and 5 non euclidean). The results are presented in table 6. As we expected computation times do not appear to differ significantly from those reported in tables 2 to 5. It should be noted that we only generated problems where the f_k's alone could not determine the depot location.

TABLE 6

FIVE 20-CITY PROBLEMS WITH FIXED COSTS ON THE DEPOTS

Problem number	type	m	First branching Strategy		Second branching Strategy	
			iterations	time	iterations	time
1	EUCLIDEAN	3	100	3.42	63	2.05
2		3	415	18.05	163	8.05
3		3	36	1.05	36	1.05
4		3	104	4.25	139	5.46
5		5	472	17.17	523	17.43
6	NON EUCLIDEAN	5	422	14.09	463	14.76
7		2	55	2.24	65	2.37
8		5	561	18.01	770	24.69
9		2	41	1.36	41	1.36
10		3	361	13.88	82	2.5

We would like to add one more comment on our results (tables 2 to 6). For a given value of n, problems tend to ve more difficult to solve when m gets larger. This was the opposite in the case of the (fixed depot) m-TSP : problems with small values of m were as a rule easier to solve as less illegal subtours were generated [6,7]. In the present problem, however, it seems that when m is larger, subtours tend to form around many depots at the same time and hence more subtour elimination constraints are required.

BIBLIOGRAPHY

[1] DEARING, P.M., FRANCIS, R.L.: "A One Facility Minimax Location Problem Involving Rectilinear Distances", ORSA Bull. 20, Sup. 2 B-469, 1972.

[2] FRANCIS, R.L., GOLDSTEIN, J.M.: "Location Theory: A Selective Bibliography", Operations Research, Vol. 22, no 4, 400-410, 1974.

[3] HANDLER, G.Y.: "Minimax Location of a Facility in an Undirected Tree Graph", Transportation Science, Vol. 7, pp.287-293.

[4] HURTER, A.P. Jr., SCHAEFFER, M.K., WENDELL, R.E.: "Solutions of Constrained Location Problems", Management Science, Vol. 22, no 1, 51-56, 1975.

[5] LAND, A.H., POWELL, S.: "Fortran Codes for Mathematical Programming", Wiley, 1973.

[6] LAPORTE, G., NOBERT, Y.: "Le problème des m-voyageurs de commerce", Rapport no 78-11, Ecole des Hautes Etudes Commerciales de Montréal, 1978.

[7] LAPORTE, G., NOBERT, Y.: "A Cutting Planes Algorithm for the m-salesman problem", submitted for publication, 1978.

[8] LAWRENCE, R.M., PENGILLY, P.J.: "The Number and Location of Depots Required for Handling Products for Distribution to Retail Stores in South-east England", Operational Research Quarterly, Vol. 20, no 1, pp.23-32, 1969.

[9] LEA, A.C.: "Location-allocation Systems: an Annotated Bibliography", Discussion Paper no 13, Department of Geography, University of Toronto, 1973.

[10] LEA, A.C.: "A Model Taxonomy and a View of Research Frontiers in Normative Locational Modelling", Paper presented to International Symposium on Locational Decisions, Banff, Alberta, April 1978.

[11] LENSTRA, J.K., RINNOOY KAN, A.H.G.: "Some Simple Applications of the Travelling Salesman Problem", Operational Research Quarterly, Vol. 26, pp.717-733, 1975.

[12] LIN, S.: "Computer Solutions of the Travelling Salesman Problem", Bell System Technical Journal, Vol. 44, pp.2245-2269, 1965.

[13] MARANZANA, F.E.: "On the Location of Supply Points to Minimize Transport Costs", Operational Research Quarterly, Vol. 15, pp. 261-270, 1964.

[14] MILIOTIS, P.: "Integer Programming Approaches to the Travelling Salesman Problem", Mathematical Programming, Vol. 10, pp. 367-378, 1976.

[15] MILIOTIS, P.: "Using Cutting Planes to Solve the Symmetric Travelling Salesman Problem", to appear in Mathematical Programming.

[16] NOBERT, Y., LAPORTE, G.: "Le problème du voyageur de commerce: revue des différentes formulations, extensions et méthodes de résolution", Rapport de recherche 78-13, Ecole des Hautes Etudes Commerciales de Montréal, 1978.

[17] PALMER, D.S.: "The Placing of Service Points to Minimize Travel", Operational Research Quarterly, Vol. 24, pp. 121-123, 1973.

[18] RAND, G.K.: "Methodological Choices in Depot Location Studies", Operational Research Quarterly, Vol. 27, pp. 241-249, 1976.

[19] ROSS, G.T., SOLAND, R.M.: "Modeling Facility Location Problems as Generalized Assignment Problems", Management Science, Vol. 24, pp. 345-357, 1977.

[20] SHAEFER, M.K., HURTER, A.P. Jr.: "An Algorithm for the Solution of a Location Problem with Metric Constraints", Naval Research Logistic Quarterly, Vol. 21, pp. 625-636, 1974.

[21] SMALLWOOD, R.D.: "Minimax Detection Station Placement", Operations Research, Vol. 13, pp. 636-646, 1965.

[22] SUSSAMS, J.E.: "Efficient Road Transport Scheduling", Gower Press, London, 1971.

[23] WREN, A., HOLLIDAY, A.: "Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Points", Operational Research Quarterly, Vol. 23, pp. 333-344, 1972.

* * *